BJUT at TREC 2015 Microblog Track: Real-Time Filtering Using Knowledge Base

Luyang Liu[1,2,3], **Zhen Yang**[1,2,3,*]
1. College of Computer Science, Beijing University of Technology, Beijing 100124, China
2. Beijing Key Laboratory of Trusted Computing, Beijing 100124, China
3. National Engineering Laboratory for CTISCP, Beijing 100124, China
*yangzhen@bjut.edu.cn

Abstract

This paper describes our efforts for TREC 2015 Microblog track. We applied the classic retrieval model combined with the external knowledge base, i.e., Wikipedia, for query expansion. Besides, we introduced the knowledge acquisition, query expansion, and retrieval model as well. Experimental results show the proposed approach is better than classical method in microblog real-time filtering with the usage of external knowledge base.

Introduction

As with the microblog such as Twitter rapidly getting popular, the information that microblog covered is rather numerous than expected. In addition, owing to the microblogs real-time property as well as its vast spreading speed, microblog have been a crucial way for users to get information through the Internet. However, faced with such colossal information, microblog search engine that designate a specific topic with respect to the users interest is the important method to satisfy users need. This years TREC 2015 is talking about the problem arose above.

TREC 2015 Microblog Track (abbreviate for MB Track) poses two involved scenarios. What we choose to take is the Scenario B: Periodic email digest. Unlike the MB Track years pasted, 2015 MB Track using the real-time tweeter data flow to dynamically evaluate the system. Whats more, this Track require participants to return the interesting microblogs with respect to the topic title and topic description that user given. The total number of the topics is 250. And each topic need to return up to 100 relevant microblogs (Duan *et al.* 2010).

First, we claw the data of Wikipedia which is thought to be relevant of each topic. Then, according to the topic information and clawed data, combining the data we get to regenerate the expanded query. Last but not the least, we remove the duplicated microblog and resort it according to its final score.

SYSTEM DESCRIPTION

During this year's MB TREC evaluation, we filtered about 10 days' tweets. Final corpus consists of 2Gbytes tweets after reducing the junk and other irrelevant tweets. Then the final result is about 210686 tweets in total 250 topics after reducing the duplicated tweets.

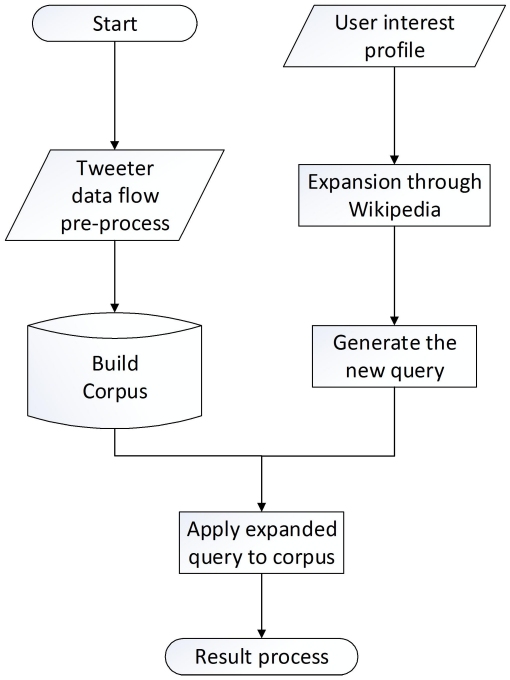

Figure 1: **System framework.**

Our system consists of 3 blocks in all: data receive and pre-handle, query expansion, result generate and re-process.

- **Tweeter data flow pre-process**
  According to the TREC official guideline and instruction, we setup a server to receive and process all the tweeter stream that evaluation needed. Note that all non-English language tweeter are supposed to be junk, thus we reduce that once we detect a non-English Unicode character in it.
- **Corpus Generation**
  When acquired the data of tweeter flow which have been pre-processed, we use the Lemur to build the corpus. We collect all day's data to generate the corpus of that day.
- **User interest profile analysis**
  TREC official user interest profile contains three main fields with the "title" containing a few keywords, the

"description" containing a one-sentence statement of the information need, and the "narrative", a paragraph-length description of the information need. We take topic title as the searching term. In addition, we take it as the key words which would be searched in Wikipedia to get the relevant information about the topic. However, given the features of Wikipedia, we simply choose limited result of retrieval through Wikipedia. At last, we take both topic description as well as the retrieval result of Wikipedia to generate the new query which would be used in the topic retrieval(Baeza-Yates *et al.* 1999).

- **Expansion through Wikipedia**
  Wikipedia which is known to all is a numerous, widespread authorized extern knowledgebase. Wikipedia is currently the world's largest knowledge resource, featuring more than three million articles. It is actively updated by tens of thousands volunteers. Each Wikipedia article describes a single topic, with a succinct, well-formed title. To get the relevant information we need, we take the topic title as the keywords to search in Wikipedia through the API. Noted that, to avoid duplication, pages that describe the same thing would be redirected to the same page, which would lead to the inadequate retrieval result. Thus, when receiving the results, we simply take no less than top 5 pages as the relevant material that we need. Accordingly, each topic has a corresponding set of extern relevant information.
- **Generate the new query**
  When acquired the relevant information set we noted above, we will take it to generate the synonym list. Then select several relevant words to regenerate the new query. According to the new query, we use it to generate Lemur Query Parameter File. Noted that we simply select the unigram Language Model with Dirichlet smoothing method as our final retrieval model(Joachims 2002)(Zhai and Lafferty 2002).
- **Apply query to the corpus and result process**
  When all the topic's Lemur Query parameter files were successfully generated, we would use it to search among the Corpus Index built before. To ensure the adequate amount of result when reduced the redundancy, the amount of retrieval result is restricted to 500 tweets. Then, these tweets would be sorted decently. Then reduce the reduplicated tweets which have similar content as well as the final score. Eventually, all result would be reformed to the format that TREC official required.

Short Texts Implicit Retrieval Framework

Some real-time extern knowledge such as Baidu, Google etc, have some defects when taken as the knowledge base. They cover too many irrelevant information such as ads or sponsors page which would import too many noise. Wikipedia, known as most famous, widespread, most authorized online encyclopedia, is a proper choose for our query expansion. Whats more, it provide convenient and fully function API for the programmer to use. Therefore, we take it as our choice.

As we described above, we assume that the each topic of interest profile $Q(T, N)$ include two parts: topic title T, topic description N. Extern relevant information $M = [m_(i, j)]$ we acquire according to the specific topic is retrieved back through Wikipedia API using topic title T. Then we generate the relevant index term-document matrix S. Noted that we simply use TF as the basic element of matrix. Whats more, all terms covered in the matrix would be filtered. After that, we use S to generate the correlation matrix which aims to find the synonym in the extern relevant information set. The correlation matrix C generated as follow:

$$M = \begin{bmatrix} m_{1,1} \cdots m_{1,d} \\ \vdots \ddots \vdots \\ m_{k,1} \cdots m_{k,d} \end{bmatrix} \tag{1}$$

$$C = M \times M^T \tag{2}$$

Then normalize $C = [c_(i, j)]$:

$$C_N = [c'_{i,j}] \tag{3}$$

$$c'_{i,j} = \frac{c_{i,j}}{c_{i,i} + c_{j,j} - c_{i,j}} \tag{4}$$

Then we could generate the synonym list now. Noted that d is number of relevant pages retrieved through Wikipedia. We sorted the index list according to its score in matrix C_N. Then select top $|d|$ index terms as the expansion terms which would combine original topic title to generate new query.

Result and Analysis

After 10 days evaluation, we filtered 210686 tweets in total. We submitted a run with its id: BJUTllyQE in this year's TREC MB Track. Its average nDCG is 0.1334.

That is what concerned after analyzing the submitted result. After analysis the result of our submission, we found that our query expansion method acts quite effectively and efficiently when concerning some topics about existed concept or events has happened long before. Some topics' nDCG even meet the official max value. However, we also found that Wiki has some defects when acting as the extern knowledge base faced with real-time filtering task. First, terms in Wikipedia were manually created through some experienced editors. Thanks to Wikipedia's great restriction towards every terms, lack of real-time would neutralize the Query expansion or even import some noises when filtering some topics about real-time events or concept. Secondly, retrieval results of Wikipedia remained quite too few to satisfy our query expansion. Thus, makes it difficult to add extra extern information to the result. Perhaps, combining some real-time knowledge base or search engines would somewhat raise the performance of our system.

CONCLUSION

In this paper, we present our systems for TREC 2015 Microblog track Real-Time Filtering Task. In the Real-Time Filtering Task, we applied a query expansion

framework utilized extern knowledge base Wikipedia. The method is common and simple, and it achieve relative good performance in some topics, while in some other topics it turned out to be relatively poor performance.

References

Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.

Yajuan Duan, Long Jiang, Tao Qin, Ming Zhou, and Heung-Yeung Shum. An empirical study on learning to rank of tweets. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 295–303. Association for Computational Linguistics, 2010.

Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM, 2002.

ChengXiang Zhai and John Lafferty. Two-stage language models for information retrieval. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 49–56. ACM, 2002.